

WHAT IS CLAIMED IS:

1 1. A method comprising:
2 converting memory access instructions in a source
3 code into a standard format;
4 generating partitions containing formatted memory
5 access instructions;
6 generating a match set, the match set including
7 matches of instruction patterns to the formatted memory access
8 instructions in the partitions; and
9 transforming the matches to vector memory access
10 instructions.

1 2. The method of claim 1 in which converting comprises
2 converting memory access instructions that read or write less
3 than a minimum data access unit (MDAU) to memory access
4 instructions that read or write a multiple of the minimum data
5 access unit.

1 3. The method of claim 2 in which converting further
2 comprises transforming the memory access instructions that
3 read or write the multiple of the minimum data access unit to
4 a format including a base address plus an offset.

1 4. The method of claim 1 in which generating partitions
2 comprises:
3 generating a data flow graph containing basic blocks
4 including the memory access instructions; and
5 for each basic block, applying a set of rules.

1 5. The method of claim 4 in which applying comprises
2 limiting a subnode of a partition to memory access
3 instructions directed to a specific memory bank.

1 6. The method of claim 5 in which applying further
2 comprises limiting the subnode of a partition to a memory read
3 or a memory write.

1 7. The method of claim 5 in which the memory bank is a
2 static random access memory (SRAM).

1 8. The method of claim 5 in which the memory bank is a
2 dynamic random access memory (DRAM).

1 9. The method of claim 5 in which the memory bank is a
2 scratchpad memory.

1 10. The method of claim 5 in which the memory back is an
2 EEPROM.

1 11. The method of claim 5 in which the memory back is
2 flash memory.

1 12. The method of claim 5 in which the memory back is a
2 NVRAM.

1 13. The method of claim 1 in which the instruction
2 patterns comprise a pattern describing instruction semantics.

1 14. The method of claim 1 in which the vector memory
2 access instructions comprise single memory access instructions
3 representing multiple memory accesses to a type of memory.

1 15. A compilation method comprising:
2 converting memory access instructions that read or
3 write less than a minimum data access unit (MDAU) to memory
4 access instructions that read or write a multiple of the
5 minimum data access unit;
6 converting the memory access instructions into a
7 format including a base address plus an offset;

8 grouping subsets of the converted memory access
9 instructions into partitions; and
10 vectorizing the converted memory access instructions
11 in the subsets that match instruction patterns.

1 16. The compilation method of claim 15 in which grouping
2 comprises:
3 generating a data flow graph containing basic blocks
4 including memory access instructions; and
5 generating subnodes in partitions, the subnodes
6 including memory access instructions directed to a memory bank
7 and performing the same operation.

1 17. The compilation method of claim 16 in which the
2 operation is a read.

1 18. The compilation method of claim 16 in which the
2 operation is a write.

1 19. The compilation method of claim 16 in which the
2 memory bank is a static random access memory (SRAM).

1 20. The compilation method of claim 16 in which the
2 memory bank is a dynamic random access memory (DRAM).

1 21. The compilation method of claim 16 in which the
2 memory bank is a scratchpad memory.

1 22. The compilation method of claim 16 in which the
2 memory bank is an EEPROM.

1 23. The compilation method of claim 16 in which the
2 memory bank is flash memory.

1 24. The compilation method of claim 16 in which the
2 memory bank is NVRAM.

1 25. The compilation method of claim 15 in which the
2 instruction patterns comprises instruction semantics.

1 26. The compilation method of claim 25 in which the
2 instruction semantics comprises segments.

1 27. A computer program product tangibly embodied in an
2 information carrier, for vectorizing memory access
3 instructions, the computer program product being operable to
4 cause data processing apparatus to:

5 convert memory access instructions residing in a source
6 code into a standard format;

7 generate partitions containing formatted memory access
8 instructions;

9 generate a match set, the match set including matches of
10 instruction patterns to the formatted memory access
11 instructions in the subsets; and

12 transform the matches to vector memory access
13 instructions.

1 28. The product of claim 27 in which converting comprises
2 converting memory access instructions that read or write less
3 than a minimum data access unit to memory access instructions
4 that read or write a multiple of the minimum data access unit.

1 29. The product of claim 28 in which converting further
2 comprises transforming the memory access instructions that
3 read or write the multiple of the minimum data access unit to
4 a format including a base address plus an offset.

1 30. The product of claim 27 in which generating
2 partitions comprises:

3 generating a data flow graph containing basic blocks
4 including memory access instructions; and

5 generating subnodes in partitions, the subnodes
6 including memory access instructions directed to a memory bank
7 and performing the same operation.